In the claims:

1. (currently amended) A method of <u>making a far call or far branch instruction using a</u> <u>near call or near branch instruction that is capable of only</u> ~~fixing link time problems~~ ~~relating to out of range limitations in~~ <u>the</u> transfer of <u>program</u> control <u>a limited</u> <u>distance from address of a near call or branch instruction to a target address</u> comprising the steps of: determining if a transfer of control is beyond a near call <u>or</u> <u>near branch</u> limitation and if so generating a link time modification of object code by ~~the~~ <u>a</u> compiler or assembler by the addition of custom generated object code or trampoline code to the link without changing the compiler generated instructions or expanding compiler generated object code for a long distance transfer of control by redirecting original call to a code which will transfer control to the ~~original~~ target address.

2. (currently amended) A method of <u>making a far call or far branch instruction using a</u> <u>near call or near branch instruction that is capable of only</u> ~~fixing link time problems~~ ~~relating to out of range limitations in~~ <u>the</u> transfer of <u>program</u> control <u>a limited</u> <u>distance from address of a branch instruction to a target address</u> comprising the steps of: determining if a transfer of control is beyond a near call <u>or near branch</u> limitation and if so generating a link time modification of object code by ~~the~~ <u>a</u> compiler or assembler by the addition of custom generated object code or trampoline code to the link <u>time object code</u> without changing the compiler generated instructions or expanding compiler generated object code for a long distance transfer of control by redirecting original call to a code which will transfer control to the ~~original~~ target address and wherein if resources are a problem a step of using a sequence of trampolines is further included.

3. (currently amended) A method of making <u>link time</u> far calls or <u>far</u> branches <u>using a</u> <u>near call or near branch instruction that is capable of only the</u> transfer of <u>program</u> <u>control a limited distance from address of a branch instruction to a target address</u> comprising the step<u>s</u> of ~~providing~~ <u>determining if a transfer of control is beyond a near</u> <u>call or near branch limitation and if so generating</u> link time modification of object

code ~~generated~~ by ~~the~~ a compiler or assembler by the addition of custom generated object code to the link without changing the compiler generated instructions or expanding compiler generated object code.

4. (currently amended) A method of branch or call instructions comprising the steps of: the compiler or assembler generating near-call instructions for all external calls, and near-return instructions for all global subroutine returns, ignoring link-time layout of sections; the linker allocating all object code sections, with no need to take into account the limitations of near-branch instructions; for each near external call C, the linker computing the distance from C to its target T and performing the following steps: determining if the call C and target T are allocated close enough to each other to permit a near call and if so, then ~~near~~ call C performs a near call to target T directly with no modification and return to consider the next call; otherwise if there is there already a trampoline S1 to target T that is linkably close enough to call C to permit a near call, then modifying call C to point to call B1 in S1 and returning to consider the next call; otherwise, creating trampoline section S1 and modifying call C to point to call B1 in S1 and add any necessary setup code to S1 and continue with following steps of determining if a second trampoline S2 is needed to reach target T and, if not, then assigning call B1 in S1 to contain a far call to target T, and return to consider the next call; otherwise, determining if a second trampoline S2 already exists to reach target T and if so, then modifying call B1 in S1 to point to existing call B2 in existing S2, and return to consider the next call; otherwise, creating a second trampoline S2 and modifying S1 to perform a far call to call B2 in S2 and add any necessary setup code to S2 and subroutine call B2 in S2 is made to contain a near call to target T and return to consider the next call.

5. (currently amended) A method of making a far call or far branch instruction using a near call or near branch instruction that is capable of only ~~fixing link-time problems relating to out-of-range branch or call instructions~~ the transfer of program control a limited distance from address of a branch instruction to a target address comprising the steps of: generating near calls at ~~the~~ a compiler or assembler for all external branches or calls; determining if the target address is too distant from a call or branch;

and if too far distant generating a trampoline section to the target <u>address</u> and re-directing the near call or branch to the trampoline section.

6. (original) The method of Claim 5 including the step of returning to an original call by returning the control through the trampoline section.

7. (currently amended) A method of <u>making a far branch or far call instruction using a near branch or near call instruction that is capable of only</u> ~~fixing link time problems relating to out of range branch or call instructions~~ <u>transferring</u> of <u>program control a limited distance from address of a branch instruction to a target address</u> comprising the steps of: computing if the target is too far distant from the <u>near</u> branch or <u>near</u> call <u>limited distance</u>; if it is too far distant then determining if there already is a trampoline section to the target <u>address</u> and if so redirect the ~~new~~ <u>far</u> call or <u>far</u> branch to that trampoline section and if there is not already a trampoline to the target <u>address</u> then generating a trampoline section to the target <u>address</u> and redirect the near call or <u>near</u> branch to the generated trampoline section.

8. (currently amended) A method of <u>making a far call or far branch instruction using a near call or near branch instruction that is capable of only</u> ~~fixing link time problems relating to out of range branch or call instructions~~ <u>transferring</u> of <u>program control a limited distance from address of a branch instruction to a target address</u> comprising the steps of: computing if the target <u>address</u> is too far distant from the <u>near</u> branch or <u>near</u> call <u>limited distance</u>; if it is too far distant then determining if there already is a trampoline section to the target <u>address</u> and if so redirect the ~~new~~ <u>far</u> call or <u>far</u> branch <u>instruction</u> to that trampoline section and if there is not already a trampoline <u>section</u> to the target <u>address</u> then generating a trampoline section to the target <u>address</u> and redirect the near call or branch to the generated trampoline section and wherein if a ~~single trampoline~~ <u>single trampoline</u> fails to work because of resources, then included are the step of: generating a second trampoline and generating a far branch or call from said first trampoline to the second trampoline section and generating at the second trampoline section a near call or branch to the ~~original~~ target <u>address</u>.

9. (currently amended) The method of Claim 8 wherein the return is a near return from the target address to the second trampoline, a far return from the second trampoline, and a near return from the first trampoline to the original call.